

WSI Zadanie 2

Zdecydowałem się na zaimplementowanie klasycznego algorytmu ewolucyjnego do rozwiązania problemu komiwojażera. Populacją w algorytmie były drogi łączące ze sobą wszystkie miasta. Populację generowałem losowo. Algorytm ewolucyjny mając określoną populację i informację o współrzędnych miast wykonywał w pętli następujące kroki przez zadaną przeze mnie liczbę iteracji:

1. Wybór połowy najlepszych kandydatów z populacji za pomocą selekcji turniejowej i stworzenie z nich nowej populacji.
2. Wykonanie dwóch mutacji na każdej z wybranych dróg polegających na podmienieniu dwóch losowych miast ze sobą (np. droga 1 -> 2 -> 3 na 1 -> 3 -> 2)
3. Wybór lepszej z dwóch mutacji (tej w której łączna droga jest krótsza) i dodanie jej do nowej populacji.
4. Zapisanie informacji o najkrótszej znalezionej drodze w aktualnej populacji.
5. Podmiana aktualnej populacji na nową.

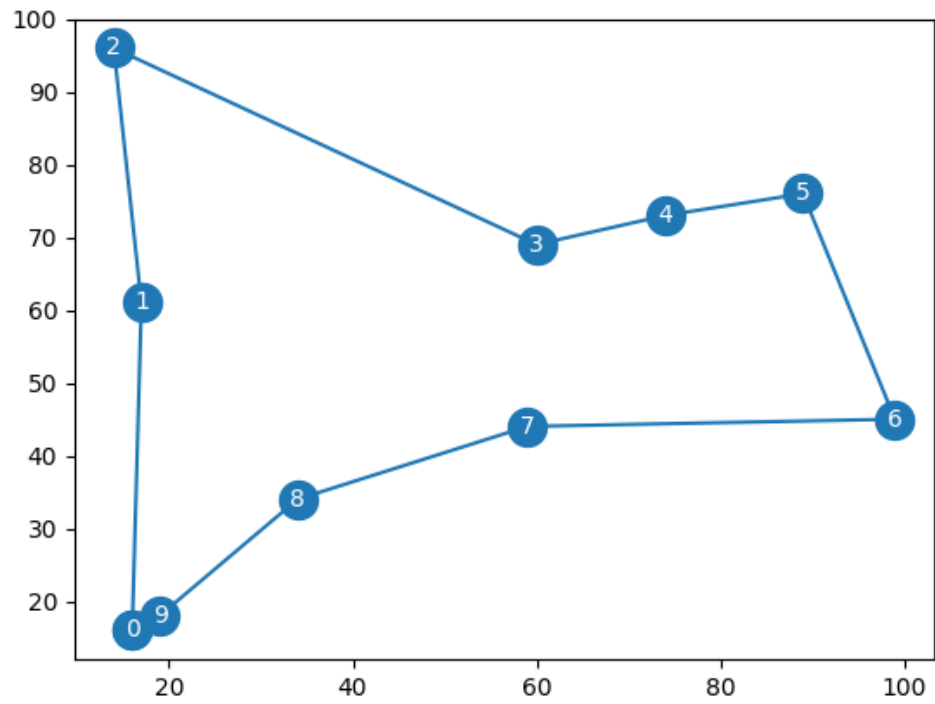
Algorytm następnie zwraca najkrótszą znaną drogę między miastami i jej długość. Użyłem modułu pyplot do stworzenia wykresów pokazujących znaną drogę, by łatwiej było się przekonać o poprawnym działaniu algorytmu.

Badanie wpływu liczności populacji na jakość rozwiązań

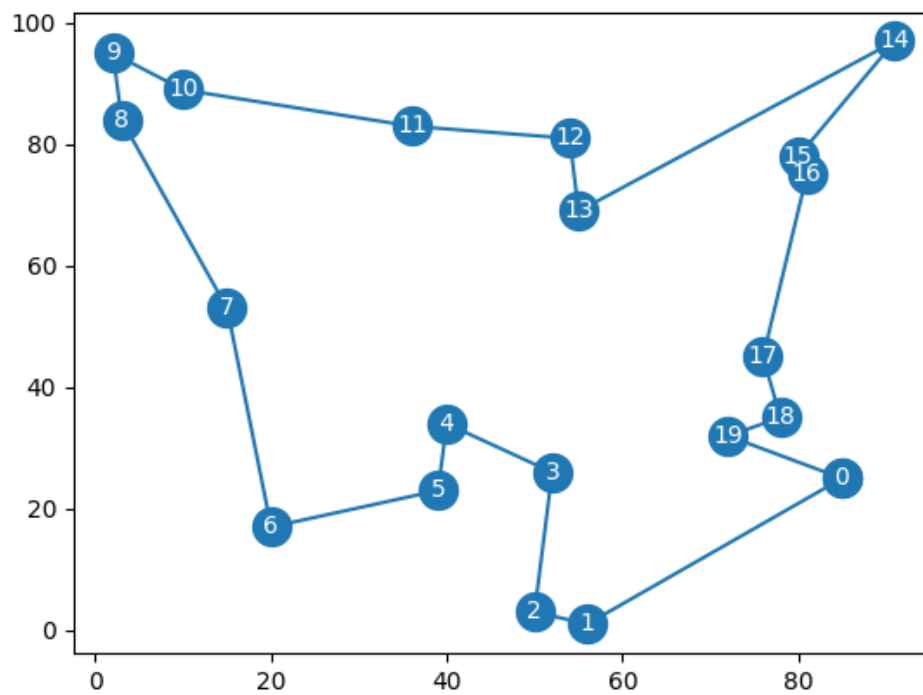
Ilość miast	Liczność populacji	Liczba iteracji algorytmu	Średnia długość najkrótszej trasy
10	50	100	288,38
10	100	100	288,38
10	50	200	288,38
20	50	250	413,24
20	100	250	405
20	50	500	398,04
30	50	500	591,62
30	100	500	581,84
30	50	1000	554,83

Na podstawie badania widać, że większa liczność populacji skutkuje na ogół nieco lepszym rozwiązaniem, ale dwukrotne zwiększenie liczby iteracji daje lepsze wyniki niż dwukrotne zwiększenie liczności populacji, mimo takiego samego wpływu na wydajność algorytmu. W moim przypadku okazało się więc, że ważniejsza od większej liczności populacji jest większa ilość obliczeń funkcji celu.

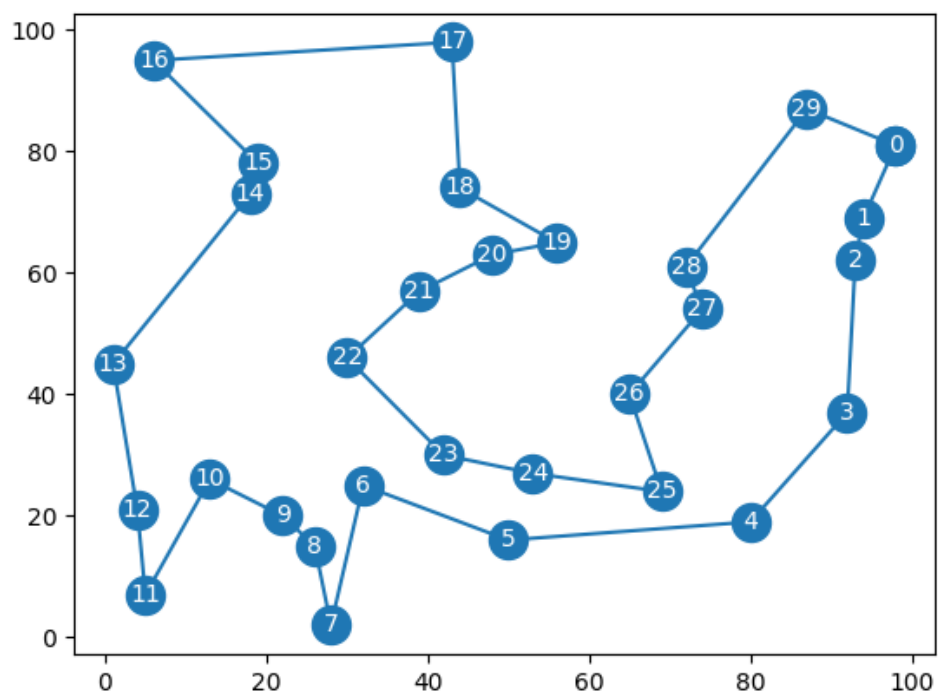
Wykresy znalezionych dróg



Wykres 1; 10 miast, 100 iteracji



Wykres 2; 20 miast, 1000 iteracji



Wykres 3; 30 miast, 5000 iteracji